REAL TIME COMMUNICATION SYSTEM FOR SPECIALLY ABLED USING AI

1.INTRODUCTION

1.1 Project Overview

Communication is a social process of exchanging information from one entity to another in verbal and non-verbal form. It defines our existence and it is an important instrument that connects people together. It comes naturally as a raw skill embedded in most people at birth and we acquired the ways of communication through cognitive learning. Communication is the basis, which drives the process of development in all the fields and it is the very core of our civilization. The ability to communicate allows us to express emotion, feelings, convey our thoughts and ideas as well as to relate our experiences. It plays an important role in the dissemination of information and sharing of knowledge especially in the academic arena. Research has found that humans started to learn how to communicate with each other since they are born not only through spoken and written languages but also body gesture, posture, facial expression and eye contacts. Communication skill might come as a natural ability in the majority of people.

1.2 Purpose

Many assistive tools or formally termed as Alternative and Augmentative Communication has been developed and employed to assist people with impaired communication skills. The term encompasses the whole combination of methods used for communication such as text to speech system, pointing gestures, facial expression and body language. Although these AACs have been widely used to assist the disabled, it is not potentially effective because most AACs are text to speech and touch screen based applications, which are unsuitable for those with severe physical abilities.

2 LITERATURE SURVEY

2.1Existing problem

There are some people afflicted with some form of physical defects which affect their ability to communicate. One of the more severe disabilities is known as “cerebral palsy”, a congenital disorder at birth which causes abnormality in their motor system. It affects their muscle movement and coordination, learning and speech abilities. Their malfunctioned motor system causes an uncontrollable and involuntary movement. They are unable to control their oral-facial muscles, thus affecting their ability to perform facial expression appropriately.

It's been over a decade since facial recognition technology has been a significant topic in the news. It made headlines in 2005 when it was used to identify the 9/11 terrorists. Ten years later, it seems that this high-tech innovation is being used less for catching criminals and more for making people feel secure.

2.2 References

[1] Prof. P.G. Ahire, K.B. Tilekary,T.A. Jawake, P.B. Warale, "Two Way Communicator between Deaf and Dumb People and Normal People", 978-1-4799-6892-3/15 31.00 c 2015 IEEE. 3

[2] Shreyashi Narayan Sawant, "Sign Language recognition System to aid Deaf- dumb People Using PCA", IJCSET ISSN : 2229-3345 Vol. 5 No. 05 May 2014.

[3] Amitkumar Shinde, Ramesh Kagalkar, "Sign Language to Text and Vice Versa Recognition using Computer Vision in Marathi", International Journal of Computer Applications (0975 – 8887) National Conference on Advances in Computing (NCAC 2015)

[4] M. Ebrahim Al-Ahdal & Nooritawati Md Tahir,'' Review in Sign Language Recognition Systems'' Symposium on Computer & Informatics(ISCI),pp:52-57, IEEE ,2012

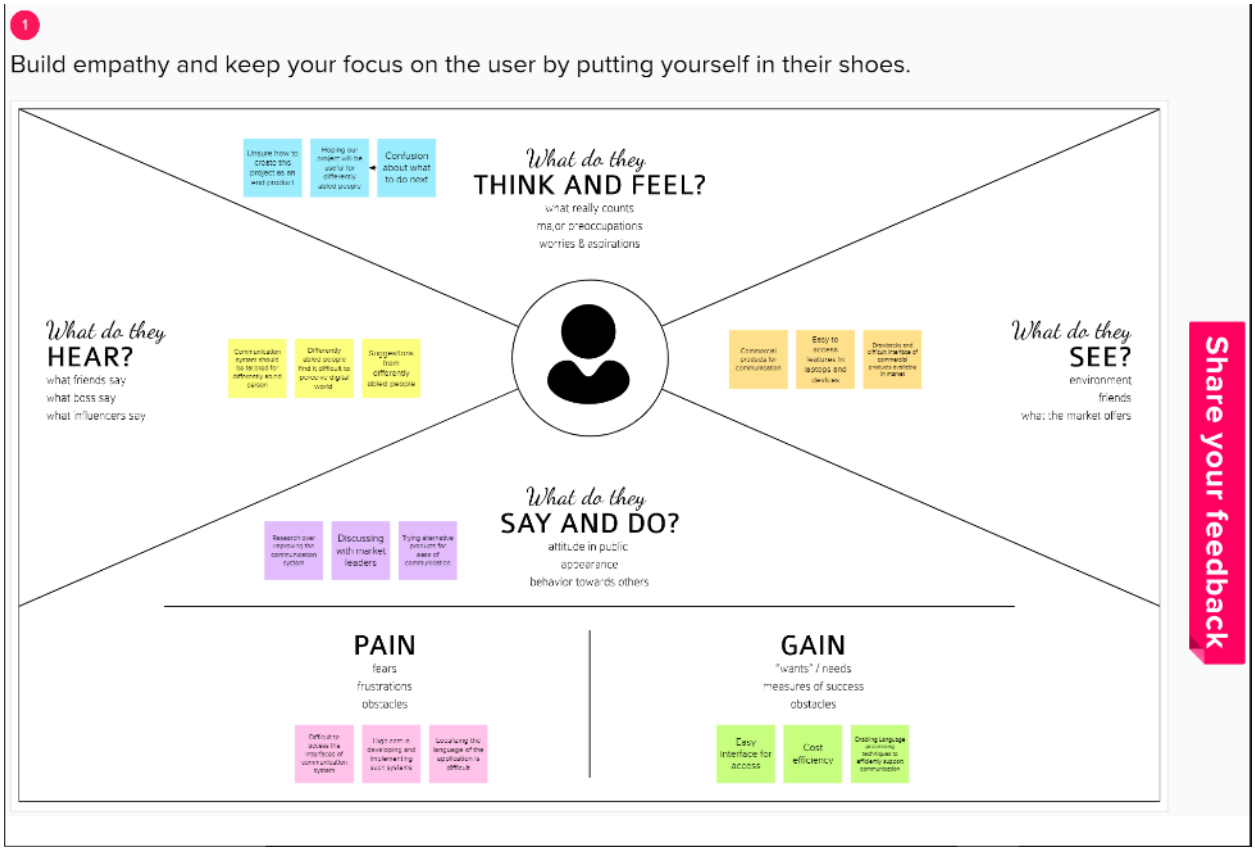
2.3 Problem Statement Definition

In our society, we have people with disabilities. The technology is developing day by day but no significant developments are undertaken for the betterment of these people. Communication between deaf-mute and a normal person has always been a challenging task. It is very difficult for mute people to convey their message to normal people. Since normal people are not trained in hand sign language. In emergency times conveying their message is very difficult. The human hand has remained a popular choice to convey information in situations where other forms like speech cannot be used. Voice Conversion System with Hand Gesture Recognition and translation will be very useful to have a proper conversation between a normal person and an impaired person in any language. The project aims to develop a system that converts the sign language into a human hearing voice in the desired language to convey a message to normal people, as well as convert speech into understandable sign language for the deaf and dumb. We are making use of a convolution neural network to create a model that is trained on different hand gestures. An app is built which uses this model. This app enables deaf and dumb people to convey their information using signs which get converted to human understandable language and speech is given as output.

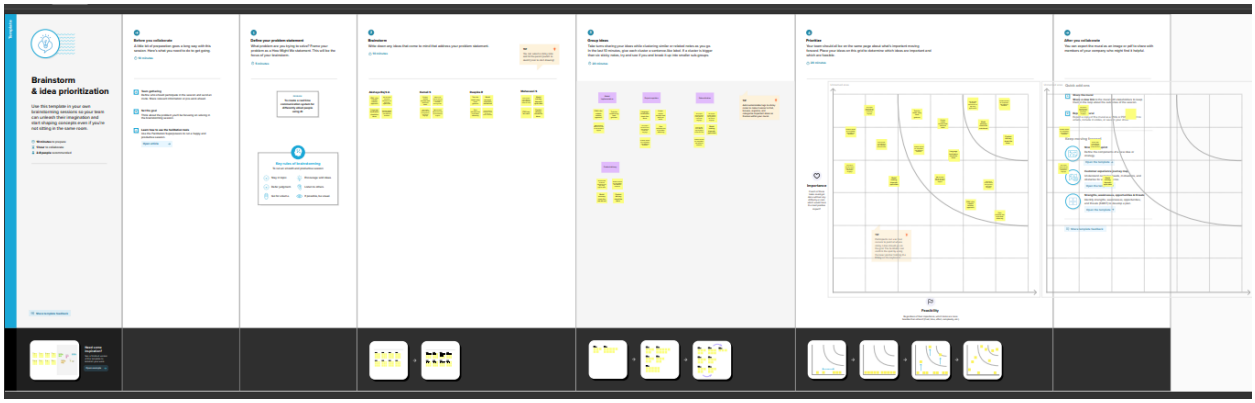
3 IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

Empathy map was done by our team using the ideas of how, why and what users think or say about their feelings over this project.



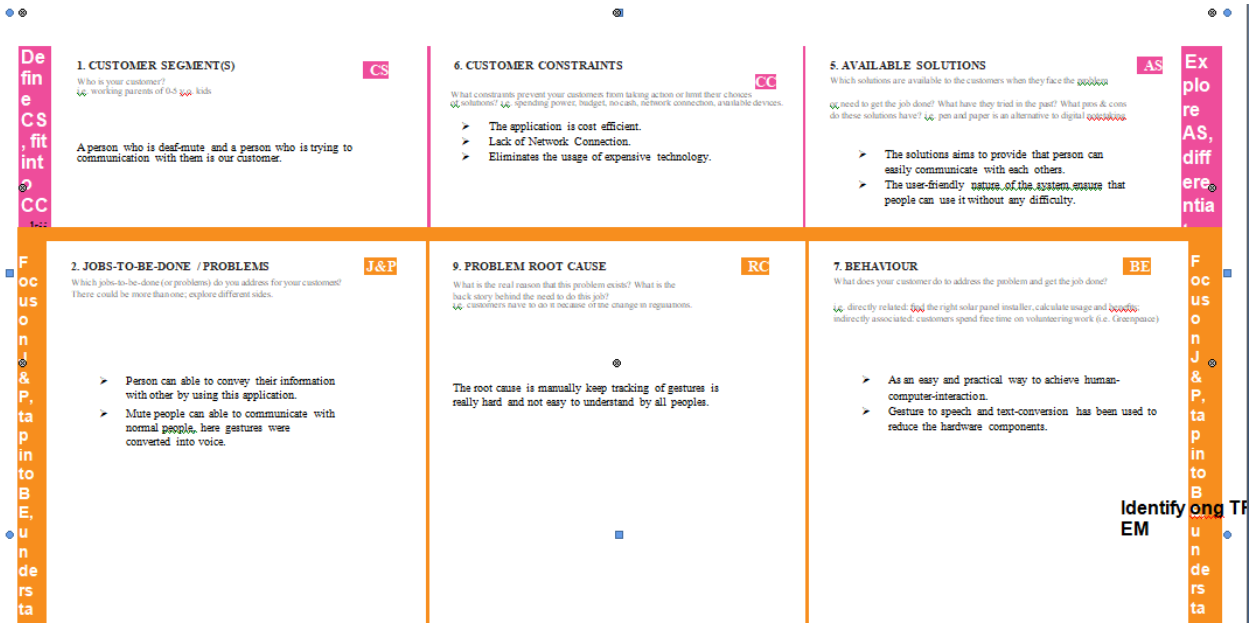
3.2 Ideation & Brainstorming

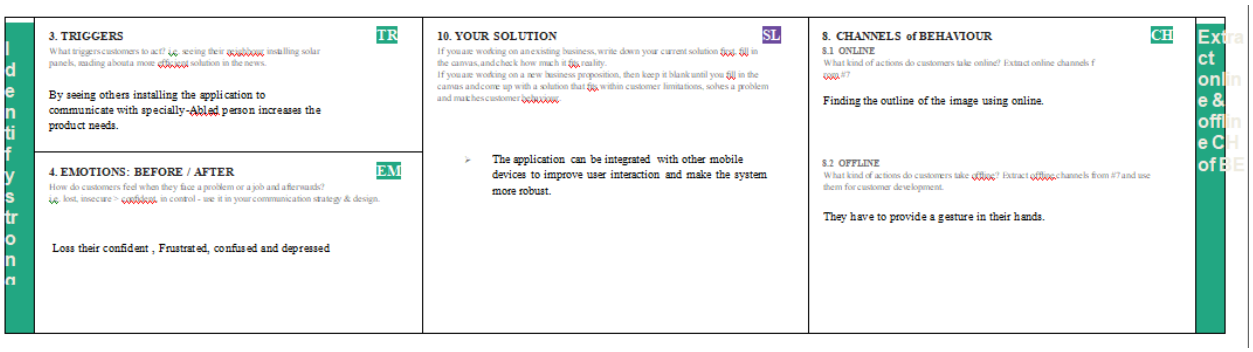


3.3 Proposed Solution

- We are making use of a convolution neural network to create a model that is trained on different hand gestures. An app is built which uses this model.
- This app enables deaf and dumb people to convey their information using signs which get converted to human-understandable language and speech is given as output.
- The technology is developing day by day but no significant developments are undertaken for the betterment of these people. Communication between deaf-mute and a normal person has always been a challenging task.

Problem Solution fit





4 REQUIREMENT ANALYSIS

4.1 Functional requirement

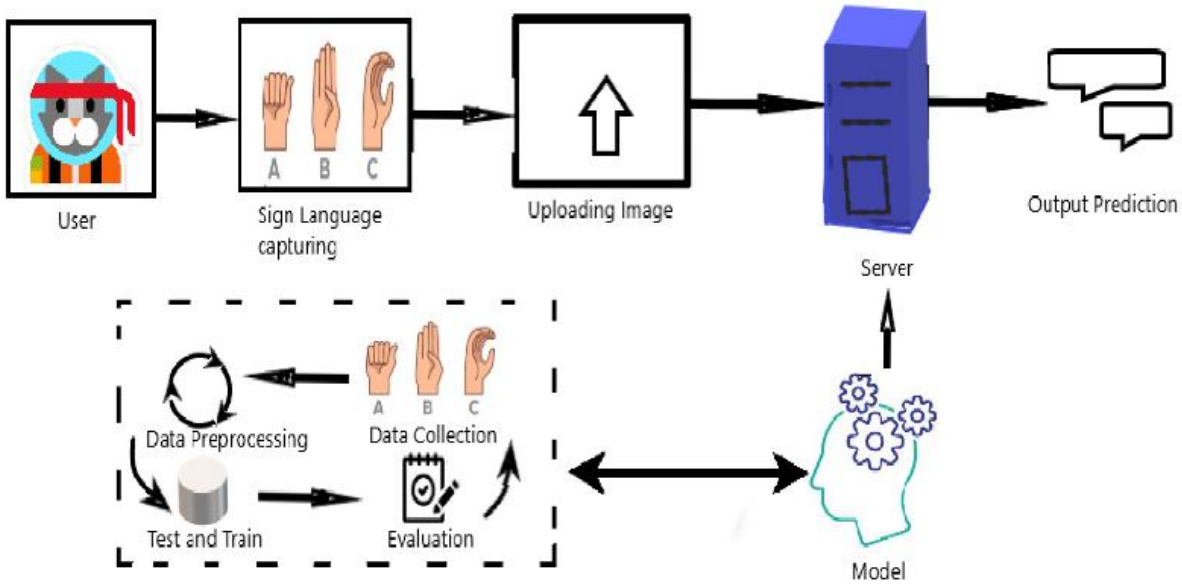
Our functional requirements are categorized as User Registration, User confirmation, Authentication, External interfaces, Transaction, Processing, Reporting and Business rules.

4.2 Non-Functional requirements

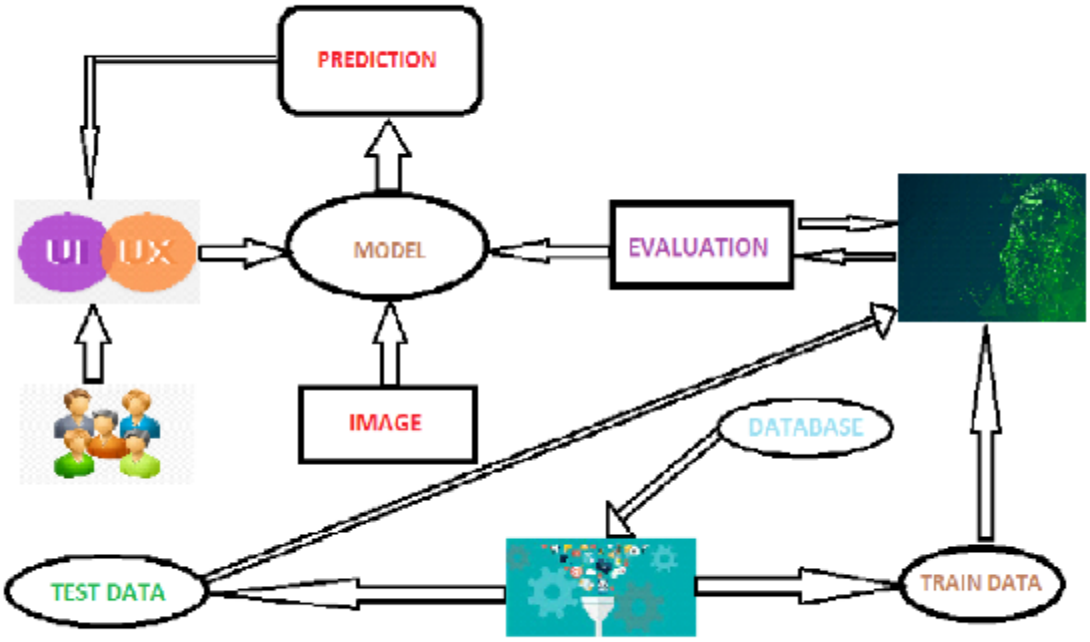
Our non-functional requirements are categorized into Usability, Security, Reliability, Performance, Availability and Scalability.

5 PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture



5.3 User Stories

- User can register for the application by entering their email, password, and confirming their password.
- User can register for the application through Google.
- User can receive confirmation content through email.
- User can log into the application by entering email & password.
- User can register by giving in their email, password and confirmation of password.
- User can log into the application by entering email & password.

6 PROJECT PLANNING & SCHEDULING

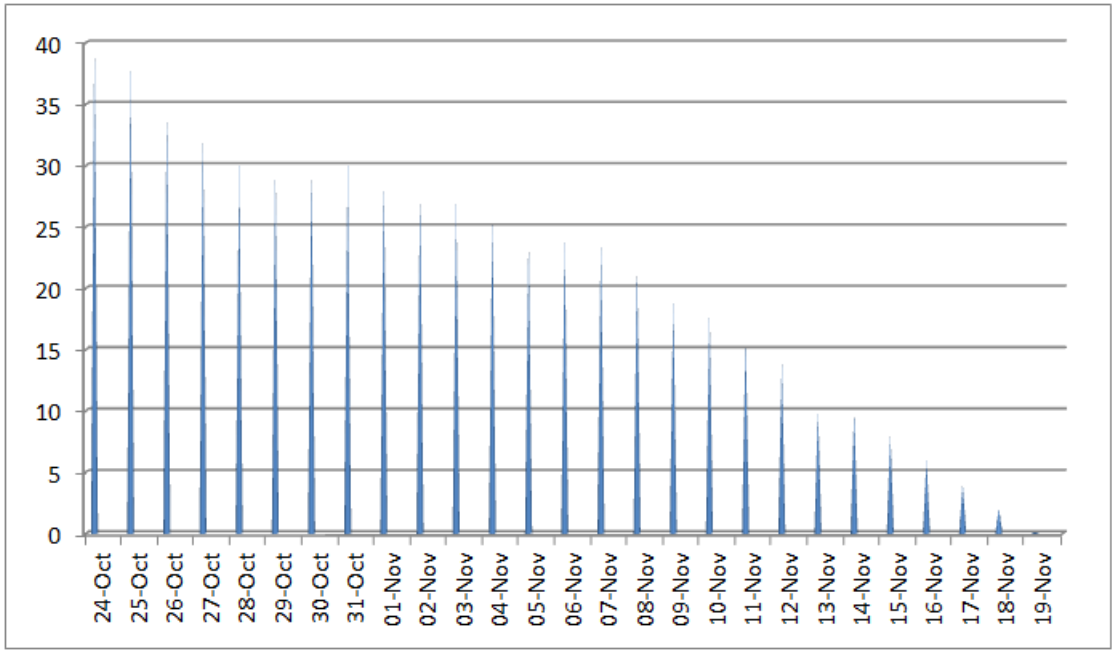
6.1 Sprint Planning & Estimation

Project Tracker, Velocity &Burn down Chart:|

Sprint	Total StoryPoints	Duration	Sprint Start Date	Sprint End Date(Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date(Actual)
Sprint-1	20	6Days	24Oct 2022	28Oct 2022	20	28Oct 2022
Sprint-2	15	6 Days	31Oct 2022	05Nov 2022	20	05Nov 2022
Sprint-3	20	7 Days	07Nov 2022	13Nov 2022	20	13Nov 2022
Sprint-4	20	6Days	14Nov 2022	19Nov 2022	20	19Nov 2022

6.2 Sprint Delivery Schedule

Burn-downChart:



7 CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

Real time video input using opencv

With OpenCV, we are capturing a video from the camera. It lets us create a video capture object which is helpful to capture videos through webcam and then we may perform desired operations on that video.

Camera.py

```
import cv2
import numpy as np
from keras.models import load_model
from keras.utils import load_img, img_to_array

class Video(object):
def __init__(self):
        self.video = cv2.VideoCapture(0)
        self.roi_start = (50, 150)
        self.roi_end = (250, 350)
        self.model = load_model('aslpng1.h5') # Execute Local Trained Model
        # self.model = load_model('IBM_Communication_Model.h5') # Execute IBM Trained
Model
```

```
        self.index=['A','B','C','D','E','F','G','H','I']
        self.y = None
    def __del__(self):
        self.video.release()
    def get_frame(self):
        ret,frame = self.video.read()
        frame = cv2.resize(frame, (640, 480))
        copy = frame.copy()
        copy = copy[150:150+200,50:50+200]
        # Prediction Start
        cv2.imwrite('image.jpg',copy)
        copy_img = load_img('image.jpg', target_size=(64,64))
        x = img_to_array(copy_img)
        x = np.expand_dims(x, axis=0)
        pred = np.argmax(self.model.predict(x), axis=1)
        self.y = pred[0]
        cv2.putText(frame,'The Predicted Alphabet is:
'+str(self.index[self.y]),(100,50),cv2.FONT_HERSHEY_SIMPLEX,1,(0,0,0),3)
        ret,jpg = cv2.imencode('.jpg', frame)
        return jpg.tobytes()
```

7.2 Feature 2

Hand Region Segmentation & Hand Detection and Tracking:

The captured images are scanned for hand gestures. This is a part of preprocessing before the image is fed to the model to obtain the prediction. The segments containing gestures are made more pronounced. This increases the chances of prediction by many folds.

The model accumulates the recognized gesture to words. The recognized words are converted into the corresponding speech. The text to speech result is a simple work around but is an invaluable feature as it gives a feel of an actual verbal conversation.

```
import
cv2
        video = cv2.VideoCapture(0)
        while True:
                ret, frame = video.read()
                cv2.imshow("Frame", frame)
                k = cv2.waitKey(1)
                if k == ord('q'):
                        break
        video.release()
        cv2.destroyAllWindows()
```

8 TESTING

8.1 Test Cases

We had manually done testing of data from the datasets that are present under the folders from A to I. The accuracy and speed of model are found by performing model tests. Here we had defined the set of test data, preconditions, expected results and post conditions, developed for a particular test scenario .

8.2 User Acceptance Testing

Figure orientation

9 RESULTS

9.1 Performance Metrics

```
from skimage.transform import resize
def detect(frame):
  img = resize(frame,(64,64,1))
  img = np.expand_dims(img,axis=0)
  if(np.max(img)>1):
    img = img/255.0
  prediction = model.predict(img)
  print(prediction)
  prediction = np.argmax(prediction,axis=1)
  print(prediction)
```

20 ADVANTAGES & DISADVANTAGES

Advantages

This app enables deaf and dumb people to convey their information using signs which get converted to human understandable language and speech is given as output.

This model builds a communication system that enables communications between deaf-dumb person and a normal person.

21 DISADVANTAGES

A disadvantage of this model is that it can only predict the letters as of now .The future scope of this model will be to predict and display words as well as sentences through gestures as this model is trained based on predicting only the letter for now.

22 CONCLUSION

The proposed communication system between Deaf and Dumb people and ordinary people are aiming for it when bridging the communication gap between two societies. Several works were done earlier in this area, but this paper adds in complete two - sided communication in an efficient manner because the system is implemented as an easily available application. So, it really serves its needs in all aspects. The above strategies prove to be efficient In terms of time and accuracy.

23 FUTURE SCOPE

The future scope of this model is be to predict and display words as well as sentences through gestures provided by differently abled people. This may enable the communication much easier than how this model could do.

24 APPENDIX

Source Code

APP.py

```
from flask import Flask, Response, render_template
from camera import Video

app = Flask(__name__)
@app.route('/')
def index():
    return render_template('index.html')

def gen(camera):
    while True:
        frame = camera.get_frame()
        yield(b'--frame\r\n'
            b'Content-Type: image/jpeg\r\n\r\n' + frame +
            b'\r\n\r\n')

@app.route('/video_feed')
def video_feed():
    video = Video()
    return Response(gen(video), mimetype='multipart/x-mixed-replace; boundary = frame')


if __name__ == '__main__':
    app.run()
```

main.py

```
import cv2

video = cv2.VideoCapture(0)

while True:
    ret, frame = video.read()
    cv2.imshow("Frame", frame)
    k = cv2.waitKey(1)
    if k == ord('q'):
            break

video.release()
cv2.destroyAllWindows()
```

GitHub & Project Demo Link